#include <stdio.h>

#include <conio.h>

#include <string.h>

#include <stdlib.h>

#include <ctype.h>

typedef struct rest

{

int id;

char nume[20];

char prenume[20];

float restanta;

char luna[10];

}restantier;

typedef struct cam

{

int id;

char nume[20];

char prenume[20];

char facultate[20];

int an;

int camera;

}camin;

typedef struct list

{

camin inf;

struct list \* urm;

}nod;

typedef struct arb

{

restantier inf;

arb\*st,\*dr;

}arbore;

char ch,ch1,ch2,ch3,luna1[20];

camin A[50];

arbore \*prim1,\*ultim,\*a1,\*a2,\*a3,\*r,\*t,\*p,\*q,\*s;

nod \*p1,\*p2,\*p3,\*prim,\*vect[50];

int i,st[100],k,n,pas,ok,z,m1,m2,n1,n2,ig,m0,aux[30],nr\_studenti,anb,v[30],ad=0,disp[30],ev,gasit,add=0,adp=0;

restantier comb[30],gol[30],pre[30],num[30];

//Fisier

void creareFisierTxt(){

FILE \*f;

char ch3;

camin a,b;

f=fopen("fis1.txt","wt"); fclose(f);

do

{

system("cls");

printf(" Creare Fisier Text \n\n");

printf("\n");

printf("Id: ");scanf("%d",&a.id);

printf("\nNume: ");scanf("%s",a.nume);

printf("\nPrenume: ");scanf("%s",a.prenume);

printf("\nFacultate: ");scanf("%s",&a.facultate);

printf("\nAn: ");scanf("%d",&a.an);

printf("\nCamera: ");scanf("%d",&a.camera);

f=fopen("fis1.txt","at");

fprintf(f,"%d %s %s %s %d %d\n",a.id,a.nume,a.prenume,a.facultate,a.an,a.camera);

fclose(f);

printf("\nMai doriti sa adaugati date?(y/n)\n");

ch3=getch();

}while (ch3!='n'&& ch3!='N');

}

void afisareFisierTxt()

{

FILE \*f;

camin a;

int pas=0;

f=fopen("fis1.txt","rt");

system("cls");

printf(" Afisare Fisier Text \n\n");

printf("\n");

printf("----------------------------------------------------------------\n");

printf("| id | nume | prenume | facultate | an | camera |\n");

printf("----------------------------------------------------------------\n");

while (!feof(f))

{

fscanf(f,"%d %s %s %s %d %d\n",&a.id,a.nume,a.prenume,a.facultate,&a.an,&a.camera);

printf("| %5d | %10s | %9s | %15s | %5d | %5d |\n",a.id,a.nume,a.prenume,a.facultate,a.an,a.camera);

pas++;

if (pas%15==0)

{

printf("\n\nApasati orice tasta pentru a continua ...\n");getch();

printf(" Afisare Fisier Text \n");

printf("\n");

printf("----------------------------------------------------------------\n");

printf("| id | nume | prenume | facultate | an | camera |\n");

printf("----------------------------------------------------------------\n");

}

}

printf("----------------------------------------------------------------\n");

printf("\nApasati orice tasta pentru a continua...\n");

fclose(f);

getch();

}

void stergereFisierTxt()

{

FILE \*f;

char ch10;

system("cls");

printf(" Stergere Fisier Text \n\n");

printf("\n");

printf("\nDoriti sa stergeti toate inregistrarile?(y/n)\n");

ch10=getch();

if (ch10=='y' || ch10=='Y')

{

f=fopen("fis1.txt","wt");

fclose(f);

printf("fisierul a fost sters \n");

}

else

printf("anulare\n");

printf("\nApasati orice tasta pentru a continua...\n");getch();

}

void adaugareFisierTxt()

{

FILE \*f;

camin a,b;

int ok=1;

system("cls");

printf(" Adaugare student in fisier Text \n\n");

printf("\n");

printf("introduceti datele noului student\n");

printf("Id: ");scanf("%d",&a.id);

printf("\nNume: ");scanf("%s",a.nume);

printf("\nPrenume: ");scanf("%s",a.prenume);

printf("\nFacultate: ");scanf("%s",a.facultate);

printf("\nAn: ");scanf("%d",&a.an);

printf("\nCamera: ");scanf("%d",&a.camera);

f=fopen("fis1.txt","rt");

while (!feof(f))

{

fscanf(f,"%d %s %s %s %d %d\n",&b.id,b.nume,b.prenume,b.facultate,&b.an,&b.camera);

if (a.id==b.id)

ok=0;

}

fclose(f);

if (ok==0)

printf("\nmai exista un student cu id %d, => nu a fost introdus in baza de date\n",a.id);

else

{

f=fopen("fis1.txt","at");

fprintf(f,"%d %s %s %s %d %d\n",a.id,a.nume,a.prenume,a.facultate,a.an,a.camera);

fclose(f);

printf("\nStudentul cu id %d a fost introdus in baza de date\n",a.id);

}

printf("\nApasati orice tasta pentru a continua....\n");

getch();

}

void stergereStudFisierTxt()

{

FILE \*f,\*f1;

camin b;

int ok=0;

system("cls");

printf(" Stergere student Fisier Text) \n");

printf("\n");

printf("Dati id-ul studentului pe care doriti sa-l stergeti: ");

scanf("%d",&k);

f=fopen("fis1.txt","rt");

f1=fopen("fis2.txt","wt");

while(!feof(f))

{

fscanf(f,"%d %s %s %s %d %d\n",&b.id,b.nume,b.prenume,b.facultate,&b.an,&b.camera);

if (k!=b.id)

fprintf(f1,"%d %s %s %s %d %d\n",b.id,b.nume,b.prenume,b.facultate,b.an,b.camera);

else

ok=1;

}

fclose(f);

fclose(f1);

if (ok)

{

remove("fis1.txt");

rename("fis2.txt","fis1.txt");

printf("\nStudentul cu id %d a fost eliminat din baza de date\n",k);

}

else

printf("\nNu exista niciun student cu id %d in baza de date\n",k);

printf("\nApasati orice tasta pentru a continua...\n");

getch();

}

void modificareStudFisierTxt()

{

FILE \*f,\*f1;

camin a,b,c;

int ok=0;

system("cls");

printf(" Modificare student Fisier Text \n");

printf("\n");

printf("Dati id studentului pe care doriti sa-l modificati: ");scanf("%d",&k);

printf("introduceti noile date ale locatarului cu id %d\n",k);

b.id=k;

printf("noul nume : ");scanf("%s",b.nume);

printf("noul prenume: ");scanf("%s",b.prenume);

printf("noua facultate : ");scanf("%s",b.facultate);

printf("noul an: ");scanf("%d",&b.an);

printf("noua camera: ");scanf("%d",&b.camera);

f=fopen("fis1.txt","rt");

f1=fopen("fis2.txt","wt");

while (!feof(f))

{

fscanf(f,"%d %s %s %s %d %d\n",&a.id,a.nume,a.prenume,a.facultate,&a.an,&a.camera);

if (k==a.id)

{

fprintf(f1,"%d %s %s %s %d %d\n",b.id,b.nume,b.prenume,b.facultate,b.an,b.camera);

ok=1;

}

else

fprintf(f1,"%d %s %s %s %d %d\n",a.id,a.nume,a.prenume,a.facultate,a.an,a.camera);

}

fclose(f);

fclose(f1);

if (ok)

{

remove("fis1.txt");

rename("fis2.txt","fis1.txt");

printf("\nmodificare reusita cu succes\n");

}

else

printf("Studentul cu id %d nu a fost gasit in baza de date\n",b.id);

printf("\nApasati orice tasta pentru a continua ...\n");

getch();

}

void FisierTxt()

{

char ch2;

pas=0;

do

{

system("cls");

printf(" MENIU FISIER TEXT\n \n");

printf("\n");

printf(" 0: Iesire\n");

printf(" 1: Creare fisier text\n");

printf(" 2: Adaugare student in fisierul text\n");

printf(" 3: Stergere student din fisierul text\n");

printf(" 4: Stergere fisier text\n");

printf(" 5: Modificare student din fisier\n");

printf(" 6: Afisare studenti din fisierul text\n");

ch2=getch();

switch(ch2)

{

case '1': creareFisierTxt();

break;

case '2': adaugareFisierTxt();

break;

case '3': stergereStudFisierTxt();

break;

case '4': stergereFisierTxt();

break;

case '5': modificareStudFisierTxt();

break;

case '6': afisareFisierTxt();

break;

}

}

while (ch2!='0');

}

//Liste simplu inlantuite

void adaugareNodLista()

{

nod \*t1,\*t2;

system("cls");

printf(" Adaugare nod Lista \n\n");

printf("\n");

t1=new nod;

printf("introduceti datele studentului!\n");

printf("Id: ");scanf("%d",&t1->inf.id);

printf("\nNume: ");scanf("%s",t1->inf.nume);

printf("\nPrenume: ");scanf("%s",t1->inf.prenume);

printf("\nFacultate: ");scanf("%s",&t1->inf.facultate);

printf("\nAn: ");scanf("%d",&t1->inf.an);

printf("\nCamera: ");scanf("%d",&t1->inf.camera);

t1->urm=NULL;

if (prim==NULL)

prim=t1;

else

{

t2=prim;

while(t2->urm!=NULL) t2=t2->urm;

t2->urm=t1;

}

printf("\nStudentul cu id %d a fost adaugat cu succes! \n",t1->inf.id);

printf("\nApasati orice tasta pentru a continua...\n");

getch();

}

void stergereNodLista()

{

nod \*t2,\*t3;

system("cls");

if(prim==NULL)

{

printf("\nTrebuie sa creati mai intai lista\n");

getch();

}

else

{

system("cls");

printf(" Stergere student Lista\n\n");

printf("\n");

printf("\nintroduceti id-ul studentului pe care doriti sa-l stergeti\n");

printf("Id: ");scanf("%d",&k);

t2=prim;

t3=prim;

while(t2->urm!=NULL && t2->inf.id!=k)

{

t3=t2;

t2=t2->urm;

}

if(t2->inf.id!=k)

printf("\nStudentul cu id %d nu a fost gasit in baza de date \n",k);

else

{

ok=1;

if (t2==prim)

{

if(t2->urm==NULL)

{

delete prim;

prim=NULL;

printf("\nLista s-a sters!\n");

}

else

{

prim=prim->urm;

delete t2;

printf("Studentul cu id %d a fost sters cu succes \n",k);

}

}

else

{

t3->urm=t2->urm;

delete t2;

printf("\nStudentul cu id %d a fost sters cu succes \n",k);

getch();

}

}

printf("\nApasati orice tasta pentru a continua...\n");

getch();

}

}

void afisareLista()

{

int pas=0;

nod\*p1;

system("cls");

printf(" Afisare Lista\n\n");

printf("\n");

printf("-----------------------------------------------------------------------------\n");

printf("| ID | Nume | Prenume | Facultate | An | Camera |\n");

printf("-----------------------------------------------------------------------------\n");

p1=prim;

while (p1)

{

pas++;

printf("| %4d | %12s | %13s | %15s | %d | %5d |\n",p1->inf.id,p1->inf.nume,p1->inf.prenume,p1->inf.facultate,p1->inf.an,p1->inf.camera);

p1=p1->urm;

if (pas%15==0)

{

printf("\nApasati orice tasta pentru a continua...\n");

getch();

pas=0;

printf(" Afisare Lista\n \n");

printf("\n");

printf("-----------------------------------------------------------------------------\n");

printf("| ID | Nume | Prenume | Facultate | An | Camera |\n");

printf("-----------------------------------------------------------------------------\n");

}

}

printf("-----------------------------------------------------------------------------\n");

printf("\nApasati orice tasta pentru a continua...\n");

getch();

}

void insertLista(nod \*p1)

{

nod \*a;

if (prim==NULL)

{

prim=new nod;

prim=p1;

}

else

{

a=prim;

while (a->urm!=NULL)

a=a->urm;

a->urm=p1;

}

}

void creareLista()

{

FILE \*f;

nod \*p1;

system("cls");

printf(" Creare Lista \n\n");

printf("\n");

prim=NULL;

f=fopen("fis1.txt","rt");

while (!feof(f))

{

p1=new nod;

p1->urm=NULL;

fscanf(f,"%d %s %s %s %d %d\n",&p1->inf.id,p1->inf.nume,p1->inf.prenume,&p1->inf.facultate,&p1->inf.an,&p1->inf.camera);

insertLista(p1);

}

fclose(f);

printf("\nlista a fost creata\n");

printf("\nApasati orice tasta pentru a continua...\n");

getch();

}

void modificareNodLista()

{

system("cls");

printf(" Modificare student Lista\n\n");

printf("\n");

printf("introduceti datele studentului pe care doriti sa-l modificati\n");

p1=new nod;

p2=new nod;

printf("Dati id-ul studentului pe care doriti sa-l modificati: ");

scanf("%d",&p1->inf.id);

printf("introduceti noile date ale studentului cu id %d\n",p1->inf.id);

printf("\nNume: ");scanf("%s",p2->inf.nume);

printf("\nPrenume: ");scanf("%s",p2->inf.prenume);

printf("\nFacultate: ");scanf("%s",&p2->inf.facultate);

printf("\nAn: ");scanf("%d",&p2->inf.an);

printf("\nCamera: ");scanf("%d",&p2->inf.camera);

p3=prim;

while ((p1->inf.id!=p3->inf.id)&&(p3->urm!=NULL))

p3=p3->urm;

if (p1->inf.id==p3->inf.id)

{

strcpy(p3->inf.nume,p2->inf.nume);

strcpy(p3->inf.prenume,p2->inf.prenume);

strcpy(p3->inf.facultate,p2->inf.facultate);

p3->inf.an=p2->inf.an;

p3->inf.camera=p2->inf.camera;

printf("\nStudentul cu id %d a fost modificat cu succes\n",p1->inf.id);

}

else printf("Studentul cu id %d nu a putut fi gasit in baza de date\n",p1->inf.id);

printf("\nApasati orice tasta pentru a continua...\n");

getch();

}

void cautareNodLista()

{

system("cls");

printf(" Cautare student Lista\n\n");

printf("\n");

printf("introduceti datele studentului pe care doriti sa-l vizualizati\n");

p1=new nod;

printf("\nIntroduceti id: ");scanf("%d",&p1->inf.id);

p2=prim;

while((p1->inf.id!=p2->inf.id)&&(p2->urm!=NULL))

p2=p2->urm;

if (p1->inf.id==p2->inf.id)

{

printf("\nStudentul cu id %d a fost gasit cu succes in baza de date\n",p1->inf.id);

printf("Nume : %s\n",p2->inf.nume);

printf("Prenume: %s\n",p2->inf.prenume);

printf("Facultate : %s\n",p2->inf.facultate);

printf("An: %d\n",p2->inf.an);

printf("Camera: %d\n",p2->inf.camera);

}

else

printf("\nStudentul cu id %d nu a fost gasit in baza de date\n\n",p1->inf.id);

printf("Apasati orice tasta pentru a continua...\n");

getch();

}

int returneaza\_nr(int x)

{

int k=0;

nod \*c;

c=prim;

while(c!=NULL)

{

if(x==c->inf.an)

k++;

c=c->urm;

}

return k;

}

nod \*returneaza\_al\_k\_student\_al\_caminului(int k, int x)

{

int i=0;

nod \*c;

c=prim;

if(x==prim->inf.an)

i++;

while((c!=NULL)&&(i<k))

{

c=c->urm;

if(x==c->inf.an)

if(i==k)

break;

else i++;

}

if(c==NULL)

return NULL;

else

return c;

}

void back(int t,int x,int nr\_studenti)

{

static int d[30], v[30];

int presp,i;

if (t < nr\_studenti)

{

for (i=0;i<nr\_studenti;i++)

if (!v[i])

{

v[i] = 1;

d[t] = i;

back(t + 1,x,nr\_studenti);

v[i] = 0;

}

}

else

{

presp=0;

for(i=0;i<nr\_studenti-1;i++)

if(returneaza\_al\_k\_student\_al\_caminului(d[i]+1,x)->inf.camera==returneaza\_al\_k\_student\_al\_caminului(d[i+1]+1,x)->inf.camera)

presp=1;

for(i=0 ; i<nr\_studenti ; i++)

printf("%s ", returneaza\_al\_k\_student\_al\_caminului(d[i]+1,x)->inf.nume);

if(presp)

printf(" Gresit\n");

else

printf(" OK\n");

}

}

void sortare\_id(int sg,int dp)

{

int i,j;

nod \*t,\*x;

i=sg;

j=dp;

x=vect[(i+j)/2];

do

{

while((i<dp)&&(vect[i]->inf.id<x->inf.id))

i++;

while((j>sg)&&(vect[j]->inf.id>x->inf.id))

j--;

if (i<=j)

{

t=vect[i];

vect[i]=vect[j];

vect[j]=t;

i++;

j--;

}

}

while(i<=j);

if (sg<j)

sortare\_id(sg,j);

if (i<dp)

sortare\_id(i,dp);

}

void quicksort()

{

char ch;

system("cls");

printf(" QUICKSORT Lista\n\n");

printf("\n");

n=0;

p1=prim;

while (p1)

{

n++;

vect[n]=new nod;

vect[n]=p1;

p1=p1->urm;

}

for (i=1; i<=n; i++)

vect[i]->urm=NULL;

sortare\_id(1,n);

prim=NULL;

for (i=1; i<=n; i++)

insertLista(vect[i]);

printf("\nLista a fost sortata!\n");

printf("\nDoriti sa o afisati?(y/n)\n");

ch=getch();

if(ch=='y')

afisareLista();

else

{

printf("\nApasati orice tasta pentru a continua...\n");

getch();

}

}

void sort(int t,int g)

{

nod \*x;

if(strcmp(vect[t]->inf.nume,vect[g]->inf.nume)>0)

{

x=vect[t];

vect[t]=vect[g];

vect[g]=x;

}

}

void interc(int m,int t,int g)

{

int i,j,k,ct;

nod \*c[30];

i=t;

j=m+1;

k=0;

while ((i<=m) && (j<=g))

{

if(strcmp(vect[i]->inf.nume,vect[j]->inf.nume)<0)

{

c[k]=vect[i];

i++;

k++;

}

else

{

c[k]=vect[j];

j++;

k++;

}

}

if(i<=m)

for(ct=i;ct<=m;ct++)

{

c[k]=vect[ct];

k++;

}

else

for(ct=j;ct<=g;ct++)

{

c[k]=vect[ct];

k++;

}

k=0;

for(i=t;i<=g;i++)

{

vect[i]=c[k];

k++;

}

}

void divimp(int t,int g)

{

int m;

if((g-t)<=1)

sort(t,g);

else

{

m=(t+g)/2;

divimp(t,m);

divimp(m+1,g);

interc(m,t,g);

}

}

void interclasare()

{

system("cls");

printf(" Interclasare(Lista) \n\n");

printf("\n");

n=0;

p1=prim;

while (p1)

{

n++;

vect[n]=new nod;

vect[n]=p1;

p1=p1->urm;

}

for (i=1; i<=n; i++)

vect[i]->urm=NULL;

divimp(1,n);

prim=NULL;

for (i=1; i<=n; i++)

insertLista(vect[i]);

printf("\nLista a fost sortata!\n");

printf("\nDoriti sa o afisati?(y/n)\n");

ch=getch();

if(ch=='y')

afisareLista();

else

{

printf("\nApasati orice tasta pentru a continua...\n");

getch();

}

}

void golire\_lista()

{

nod \*p1;

FILE \*g;

system("cls");

printf(" Golire lista in Fisier Text\n\n");

g=fopen("fis4.txt","at");

p1=prim;

while(p1!=NULL)

{

fprintf(g,"%d %s %s %s %d %d\n",p1->inf.id,p1->inf.nume,p1->inf.prenume,p1->inf.facultate,p1->inf.an,p1->inf.camera);

p1=p1->urm;

}

fclose(g);

remove("fis1.txt");

rename("fis4.txt","fis1.txt");

printf("\nLista a fost golita in fisier\n");

printf("\nApasati orice tasta pentru a continua...\n");

getch();

}

void Liste()

{

char ch;

do

{

system("cls");

printf(" MENIU LISTE\n\n");

printf("\n");

printf(" 0: Iesire\n");

printf(" 1: Creare lista\n");

if(prim!=NULL)

{

printf(" 2: Adaugare student in lista\n");

printf(" 3: Stergere student din lista\n");

printf(" 4: Modificarea unui student din lista\n");

printf(" 5: Cautarea unui nod in lista\n");

printf(" 6: Afisare lista camin\n");

printf(" 7: Permutari studenti an\n");

printf(" 8: Sortare dupa id-quicksort\n");

printf(" 9: Sortare dupa nume-interclasare\n");

printf(" A: Golire lista in fisier txt\n");

}

ch=getch();

switch (ch)

{

case '1':creareLista();

break;

case '2':adaugareNodLista();

break;

case '3':stergereNodLista();

break;

case '4':modificareNodLista();

break;

case '5':cautareNodLista();

break;

case '6':afisareLista();

break;

case '7':system("cls");

printf(" Permutari studenti/an\n\n");

printf("\nIntroduceti anul : ");

scanf("%d",&anb);

nr\_studenti=returneaza\_nr(anb);

if(nr\_studenti==0)

{

printf("\nNu am gasit studenti din anul %d in baza de date!\n",anb);

printf("\nApasati orice tasta pentru a continua...\n");

getch();

break;

}

else

{

printf("Am gasit %d studenti in anul %d, apasati o tasta pt a le permuta\n",nr\_studenti,anb);

getch();

back(0,anb,nr\_studenti);

getch();

break;

}

case '8':quicksort();

break;

case '9':interclasare();

break;

case 'a':

case 'A': golire\_lista();

break;

}

}

while (ch!='0');

}

//Fisier binar

void creareFisierBin()

{

FILE \*g;

char ch3;

restantier a,b;

g=fopen("fis2.bin","wb");

fclose(g);

g=fopen("fis2.bin","ab");

fseek(g,0,SEEK\_END);

do

{

system("cls");

printf(" Creare Fisier Binar \n\n");

printf("\n");

printf("Id: ");scanf("%d",&a.id);

printf("\nNume: ");scanf("%s",a.nume);

printf("\nPrenume: ");scanf("%s",a.prenume);

printf("\nRestanta: ");scanf("%f",&a.restanta);

printf("\nLuna: ");scanf("%s",a.luna);

fwrite(&a,sizeof(rest),1,g);

printf("\nMai doriti sa adaugati date?(y/n)\n");

ch3=getch();

}

while (ch3!='n'&& ch3!='N');

fclose(g);

printf("\nApasati orice tasta pentru a continua...\n");

getch();

}

void afisareFisierBin()

{

FILE \*g;

int k=0,pas=0,cont;

restantier a;

g=fopen("fis2.bin","rb");

system("cls");

fseek(g,0,SEEK\_SET);

printf(" Afisare Fisier Binar\n \n");

printf("\n");

printf("------------------------------------------------------------------ \n");

printf("| ID | Nume | Prenume | Restanta | Luna | \n");

printf("------------------------------------------------------------------\n");

while(!feof(g))

{

fread(&a,sizeof(rest),1,g);

k++;

}

cont=0;

fseek(g,0,SEEK\_SET);

while (!feof(g) && cont<k-1)

{

fread(&a,sizeof(rest),1,g);

cont++;

printf("| %5d | %14s | %11s | %10.2f | %10s | \n",a.id,a.nume,a.prenume,a.restanta,a.luna);

pas++;

if (pas%15==0)

{

printf("\n\nApasati orice tasta pentru a continua ...\n");

getch();

printf(" Afisare Fisier Binar\n \n");

printf("\n");

printf("--------------------------------------------------------------\n");

printf("| ID | Nume | Prenume | Restanta | Luna | \n");

printf("--------------------------------------------------------------\n");

}

}

printf("-----------------------------------------------------------------\n");

printf("\nApasati orice tasta pentru a continua...\n");

fclose(g);

getch();

}

void stergereFisierBin()

{

FILE \*g;

char ch;

system("cls");

printf(" Stergere Fisier Binar\n \n");

printf("\n");

printf("\nDoriti sa stergeti toate inregistrarile?(y/n)\n");

ch=getch();

if (ch=='y' || ch=='Y')

{

g=fopen("fis2.Bin","wb");

fclose(g);

printf("\nfisierul a fost sters \n");

}

else

printf("\nanulare\n");

printf("\nApasati orice tasta pentru a continua...\n");getch();

}

void adaugareFisierBin()

{

FILE \*g;

restantier a,b;

int ok=1;

system("cls");

printf(" Adaugare student Fisier Binar\n \n");

printf("\n");

printf("introduceti datele noului student\n");

printf("Id: ");scanf("%d",&a.id);

printf("\nNume: ");scanf("%s",a.nume);

printf("\nPrenume: ");scanf("%s",a.prenume);

printf("\nRestanta: ");scanf("%f",&a.restanta);

printf("\nLuna: ");scanf("%s",a.luna);

g=fopen("fis2.bin","rb");

while (!feof(g))

{

fread(&b,sizeof(rest),1,g);

if (a.id==b.id)

ok=0;

}

fclose(g);

if (ok==0)

printf("\nmai exista un student cu id %d, => nu a fost introdus in baza de date\n",a.id);

else

{

g=fopen("fis2.bin","ab");

fwrite(&a,sizeof(rest),1,g);

fclose(g);

printf("\nStudentul cu id %d a fost introdus in baza de date\n",a.id);

}

printf("\nApasati orice tasta pentru a continua\n");

getch();

}

void stergereStudFisierBin()

{

FILE \*g,\*g1;

restantier b;

int ok=0,t=0,cont;

system("cls");

printf(" Stergere student Fisier Binar\n \n");

printf("\n");

printf("Dati id studentului pe care doriti sa-l stergeti: ");scanf("%d",&k);

g=fopen("fis2.bin","rb");

g1=fopen("fis3.bin","wb");

t=0;

while(!feof(g))

{

fread(&b,sizeof(rest),1,g);

t++;

}

fseek(g,0,SEEK\_SET);

cont=0;

while(!feof(g) && cont<t-1)

{

fread(&b,sizeof(rest),1,g);

cont++;

if (k!=b.id)

{

fwrite(&b,sizeof(rest),1,g1);

}

else

ok=1;

}

fclose(g);

fclose(g1);

if (ok)

{

remove("fis2.bin");

rename("fis3.bin","fis2.bin");

printf("\nStudentul cu id %d a fost eliminat din baza de date\n",k);

}

else

printf("\nNu exista niciun student cu id %d in baza de date\n",k);

printf("\nApasati orice tasta pentru a continua...\n");

getch();

}

void modificareStudFisierBin()

{

FILE \*g,\*g1;

restantier a,b,c;

int ok=0,t=0,cont=0;

system("cls");

printf(" Modificare Fisier Binar \n\n");

printf("\n");

printf("Dati id studentului pe care doriti sa-l modificati: ");scanf("%d",&k);

printf("introduceti noile date ale studentului cu id %d\n",k);

b.id=k;

printf("noul nume : ");scanf("%s",b.nume);

printf("noul prenume: ");scanf("%s",b.prenume);

printf("noua restanta : ");scanf("%f",&b.restanta);

printf("noua Luna: ");scanf("%s",b.luna);

g=fopen("fis2.bin","rb");

g1=fopen("fis3.bin","wb");

fseek(g1,0,SEEK\_SET);

while(!feof(g))

{

fread(&a,sizeof(rest),1,g);

t++;

}

fseek(g,0,SEEK\_SET);

while (!feof(g) && cont<t-1)

{

fread(&a,sizeof(rest),1,g);

cont++;

if (k==a.id)

{

fwrite(&b,sizeof(rest),1,g1);

ok=1;

}

else

fwrite(&a,sizeof(rest),1,g1);

}

fseek(g1,0,SEEK\_SET);

fclose(g); fclose(g1);

if (ok)

{

remove("fis2.bin");

rename("fis3.bin","fis2.bin");

printf("modificare reusita cu succes\n");

}

else

printf("\nStudentul cu id %d nu a fost gasit in baza de date\n",b.id);

printf("\nApasati orice tasta pentru a continua ...\n");

getch();

}

void FisierBin()

{

char ch2;

FILE \*g;

rest a;

pas=0;

do

{

system("cls");

printf(" MENIU FISIERE BINARE\n \n");

printf("\n");

printf(" 0: Iesire\n");

printf(" 1: Creare fisier binar\n");

printf(" 2: Adaugare student in fisierul binar\n");

printf(" 3: Stergere student din fisierul binar\n");

printf(" 4: Stergere fisier binar\n");

printf(" 5: Modificare student din fisier binar\n");

printf(" 6: Afisare studenti din fisier binar\n");

ch2=getch();

switch(ch2)

{

case '1': creareFisierBin();

break;

case '2': adaugareFisierBin();

break;

case '3': stergereStudFisierBin();

break;

case '4': stergereFisierBin();

break;

case '5': modificareStudFisierBin();

break;

case '6': afisareFisierBin();

break;

}

}

while (ch2!='0');

}

//Arbori

void Inordine1(arbore \*a1)

{

if (a1!=NULL)

{

Inordine1(a1->st);

printf("%d %s %s %.2f %s \n",a1->inf.id,a1->inf.nume,a1->inf.prenume,a1->inf.restanta,a1->inf.luna);

Inordine1(a1->dr);

}

}

void Preordine1(arbore \*a1)

{

if (a1!=NULL)

{

printf("%d %s %s %.2f %s \n",a1->inf.id,a1->inf.nume,a1->inf.prenume,a1->inf.restanta,a1->inf.luna);

Preordine1(a1->st);

Preordine1(a1->dr);

}

}

void Postordine1(arbore \*a1)

{

if (a1!=NULL)

{

Postordine1(a1->st);

Postordine1(a1->dr);

printf("%d %s %s %.2f %s \n",a1->inf.id,a1->inf.nume,a1->inf.prenume,a1->inf.restanta,a1->inf.luna);

}

}

void afisare\_arbore(int n,arbore \*a1)

{

int i;

if (a1)

{

afisare\_arbore(n+7,a1->dr);

for (i=1; i<=n; i++) printf(" ");

printf("%d\n",a1->inf.id);

for (i=1; i<=n; i++) printf(" ");

printf("\n");

afisare\_arbore(n+7,a1->st);

}

}

arbore\* AdaugareInArb(arbore \*arb,restantier x)

{

if(arb==NULL)

{

arb=new arbore;

arb->st=NULL;

arb->dr=NULL;

arb->inf=x;

}

else if(arb->inf.id<x.id)

{

arb->dr= AdaugareInArb(arb->dr,x);

}

else if(arb->inf.id>x.id)

{

arb->st=AdaugareInArb(arb->st,x);

}

else

printf("\nExista nodul cu informatia %d\n",x.id);

return arb;

}

arbore \* creareArb(arbore \*arb)

{

restantier x;

FILE \*g;

int t=0,cont=0;

system("cls");

arb=NULL;

g=fopen("fis2.bin","rb");

while(!feof(g))

{

fread(&x,sizeof(rest),1,g);

t++;

}

cont=0;

fseek(g,0,SEEK\_SET);

do

{

fread(&x,sizeof(rest),1,g);

cont++;

if(x.id>0)

arb=AdaugareInArb(arb,x);

}

while( cont<t-1 && !feof(g));

fclose(g);

printf("\nArborele a fost creat\n");

printf("\nApasati orice tasta pentru a continua....\n");

getch();

return arb;

}

void inserare\_nod\_arb()

{

arbore \*a;

system("cls");

printf(" Inserare Arbori\n \n");

printf("\n");

printf("introduceti datele noului student: \n");

a=new arbore;

printf("\nId: ");scanf("%d",&a->inf.id);

printf("\nNume: ");scanf("%s",a->inf.nume);

printf("\nPrenume: ");scanf("%s",a->inf.prenume);

printf("\nRestanta: ");scanf("%f",&a->inf.restanta);

printf("\nLuna: ");scanf("%s",a->inf.luna);

a->st=NULL;

a->dr=NULL;

s=AdaugareInArb(prim1,a->inf);

}

void gaseste(arbore \*a1,arbore \*a2)

{

if (a1)

{

if (a1->st==a2 || a1->dr==a2)

a2=a1;

else

{

gaseste(a1->st,a2);

gaseste(a1->dr,a2);

}

}

}

void Elimina()

{

if(t->dr!=NULL)

{

r=t;

t=t->dr;

Elimina();

}

else

{

q->inf.id=t->inf.id;

if((r!=NULL)&&(r->dr==t))

r->dr=t->st;

else

q->st=t->st;

q=t;

}

}

void Stergere(int nr)

{

if(p==NULL)

printf("\n Inregistrarea %d nu exista\n",nr);

else

if(nr<p->inf.id)

{

r=p;

p=p->st;

Stergere(nr);

}

else

if(nr>p->inf.id)

{

r=p;

p=p->dr;

Stergere(nr);

}

else

{

q=p;

if (q==prim1)

{

if (prim1->st==NULL)

prim1=prim1->dr;

else

if (prim1->dr==NULL)

prim1=prim1->st;

else

{

t=q->st;

Elimina();

}

}

else

if (q->dr==NULL)

{

if(r->st==p)

r->st=q->st;

else

r->dr=q->st;

}

else

if(q->st==NULL)

{

if(r->st==p)

r->st=q->dr;

else

r->dr=q->dr;

}

else

{

t=q->st;

Elimina();

}

delete q;

printf("\nInregistrarea %d a fost stearsa!\n",nr);

getch();

}

}

void stergereNodArbore()

{

int nr1,i;

system("cls");

printf(" Stergere Arbori \n \n");

printf("\n");

printf("\nintroduceti datele studentului pe care doriti sa-l stergeti\n");

a1=new arbore;

printf("Dati id: ");scanf("%d",&nr1);

ok=0;

p=prim1;

Stergere(nr1);

printf("\nApasati orice tasta pentru a continua...\n");

getch();

}

void Inordine2(arbore \*a1)

{

int pas=0,i=0;

if (a1!=NULL)

{

Inordine2(a1->st);

ad++;

comb[ad]=a1->inf;

Inordine2(a1->dr);

}

}

void INIT()

{

int i;

for(i=0;i<=ad;i++)

v[i]=0;

ig=1;

}

void GEN\_SUBM()

{

int i;

i=ad;

while(v[i]==1)

{

v[i]=0;

i--;

}

if(i==0)

ig=0;

else

v[i]=1;

}

void PREL()

{

int i,s,ok=0;

char chh;

for(i=1;i<=ad;i++)

if(v[i]!=0)

ok=1;

if(ok)

{

for(i=1;i<=ad;i++)

if(v[i]==1)

printf("%12s ",comb[i].nume);

for(i=1;i<=ad;i++)

if(v[i]==1)

if(strcmp(comb[i].luna,luna1))

s=0;

else s=1;

if(s)

printf("%5s\n","ok");

else

printf("%5s\n","Nu");

}

}

void Postordine2(arbore \*a1)

{

if (a1!=NULL)

{

Postordine2(a1->st);

Postordine2(a1->dr);

add++;

gol[add]=a1->inf;

}

}

void Preordine2(arbore \*a1)

{

if (a1!=NULL)

{

adp++;

pre[adp]=a1->inf;

Preordine2(a1->st);

Preordine2(a1->dr);

}

}

void golire\_arbore()

{

int i;

nod \*p1;

FILE \*f4;

char ch;

ad=0;add=0;adp=0;

system("cls");

printf(" Golire Arbore in Fisier Binar\n\n");

f4=fopen("fis4.bin","ab");

printf("\nIn ce ordine doriti sa goliti arborele in fisier?:\n1-Inordine\n2-Preordine\n3-Postordine\n");

ch=getch();

switch(ch)

{

case '1':ad=0;Inordine2(prim1);

for(i=1;i<=ad;i++)

fwrite(&comb[i],sizeof(rest),1,f4);

fclose(f4);

break;

case '2':adp=0;Preordine2(prim1);

for(i=1;i<=adp;i++)

fwrite(&pre[i],sizeof(rest),1,f4);

fclose(f4);

break;

case '3':add=0;Postordine2(prim1);

for(i=1;i<=add;i++)

fwrite(&gol[i],sizeof(rest),1,f4);

fclose(f4);

break;

}

remove("fis2.bin");

rename("fis4.bin","fis2.bin");

printf("\nArborele a fost golit in fisier\n");

printf("\nApasati orice tasta pentru a continua...\n");

getch();

}

void init()

{

st[k]=0;

}

int solutie(int k)

{

if (k==ad)

return 1;

else

return 0;

}

void succesor(int k)

{

if (st[k]<ad)

{

ok=1;

st[k]=st[k]+1;

}

else

ok=0;

}

void tipar()

{

for(i=1; i<=ad; i++)

printf("%s ",comb[st[i]].nume);

printf("\n");

}

void valid()

{

int i;

gasit=1;

for (i=1; i<k; i++)

if (st[i]==st[k] || strcmp(comb[st[i]].luna,comb[st[k]].nume)==0)

gasit=0;

}

void permutari()

{

char chh;

system("cls");

printf(" Permutari Arbore\n\n");

printf("\n");

ad=0;

Inordine2(prim1);

k=1;

init();

while (k>0)

{

if (st[k]<ad)

{

succesor(k);

if (ok)

valid();

if (gasit && ok)

if (solutie(k))

{

tipar();

printf("\nMai doriti o alta permutare?y/n\n\n");

chh=getch();

if (chh=='y')

{

}

else

{

printf("\nApasati orice tasta pentru a continua...\n");

getch();

break;

}

}

else

{

k++;

init();

}

}

else

k--;

}

}

void sortare\_numarare()

{

int i,j;

char ch;

ad=0;

Inordine2(prim1);

for(i=1;i<=ad;i++)

aux[i]=1;

for(j=2;j<=ad;j++)

for(i=1;i<j;i++)

if(comb[i].restanta<=comb[j].restanta)

aux[j]++;

else

aux[i]++;

for(i=1;i<=ad;i++)

num[aux[i]]=comb[i];

for(i=1;i<=ad;i++)

comb[i]=num[i];

system("cls");

printf("\nInformatiile din arbore au fost sortate!\n");

printf("\nDoriti sa le afisati?(y/n)\n");

ch=getch();

if(ch=='y' || ch=='Y')

{

for(i=1;i<=ad;i++)

printf("%5d %12s %12s %5.2f %12s\n",comb[i].id,comb[i].nume,comb[i].prenume,comb[i].restanta,comb[i].luna);

printf("\nApasati orice tasta pentru a continua...\n");

getch();

}

else

{

printf("\nApasati orice tasta pentru a continua...\n");

getch();

}

}

void Arbori()

{

char ch2,chh;

do

{

system("cls");

printf(" MENIU ARBORI \n");

printf("\n");

printf(" 0: Iesire\n");

printf(" 1: Creare arbore\n");

if(prim1!=NULL)

{

printf(" 2: Afisare aborescenta\n");

printf(" 3: Inserare nod\n");

printf(" 4: Stergere nod\n");

printf(" 5: Afisare inordine\n");

printf(" 6: Afisare preordine\n");

printf(" 7: Afisare postordine\n");

printf(" 8: Afisarea grupurilor de studenti-restanta luna x\n");

printf(" 9: Golire arbore in fisier\n");

printf(" A: Permutari studenti-backtraking\n");

printf(" B: Sortare dupa suma restanta-numarare\n");

}

ch2=getch();

switch(ch2)

{

case '1': prim1=creareArb(prim1);

break;

case '2': system("cls");

printf(" Afisare Arbore\n\n");

afisare\_arbore(0,prim1);

printf("\nApasati orice tasta pentru a continua...\n");

getch();

break;

case '3': inserare\_nod\_arb();

break;

case '4': stergereNodArbore();

break;

case '5': system("cls");

printf(" Afisare INORDINE\n\n");

Inordine1(prim1);

printf("\nApasati orice tasta pentru a continua...\n");

getch();

break;

case '6': system("cls");

printf(" Afisare PREORDINE\n\n");

Preordine1(prim1);

printf("\nApasati orice tasta pentru a continua...\n");

getch();

break;

case '7': system("cls");

printf(" Afisare POSTORDINE\n\n");

Postordine1(prim1);

printf("\nApasati orice tasta pentru a continua...\n");

getch();

break;

case '8': system("cls");

printf(" Grupuri studenti\n\n");

printf("\nDati luna pt care sa se scrie grupurile\n");

scanf("%s",luna1);ad=0;

Inordine2(prim1);

INIT();

while(ig)

{

PREL();

GEN\_SUBM();

printf("\nMai doriti sa vizualizati alt grup?(y/n)\n");

chh=getch();

if(chh=='y' || chh=='Y')

{

}

else

{

printf("\nApasati orice tasta pentru a continua...\n");

getch();

break;

}

}

break;

case '9': golire\_arbore();

break;

case 'a':

case 'A': permutari();

break;

case 'b':

case 'B': sortare\_numarare();

break;

}

}

while(ch2!='0');

}

void main()

{

char ch;

do

{

system("cls");

printf(" MENIU PRINCIPAL\n\n");

printf("\n\n");

printf("0: Iesire\n\n");

printf("1: Fisier Text\n\n");

printf("2: Liste\n\n");

printf("3: Arbori\n\n");

printf("4: Fisier binar\n\n");

ch=getch();

switch (ch)

{

case '1': FisierTxt();

break;

case '2': Liste();

break;

case '3': Arbori();

break;

case '4': FisierBin();

break;

}

}

while (ch!='0');

}